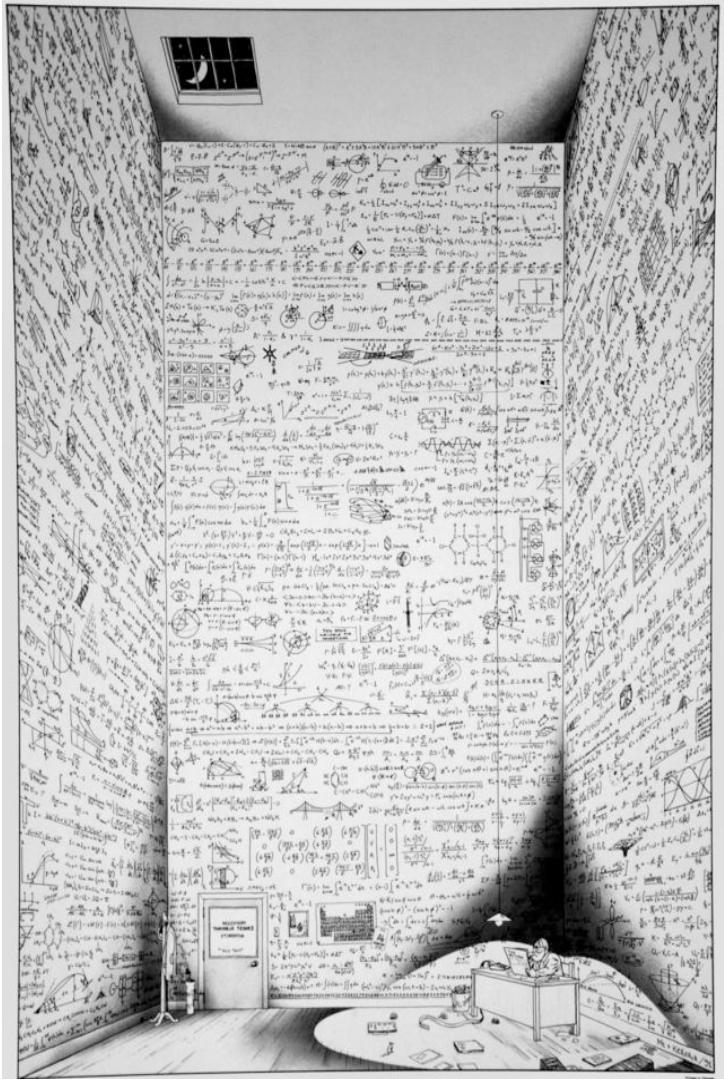
Towards Human Level Cognition with Stochastic Bayesian Computing

Yangdong Steve Deng
School of Software
Tsinghua University

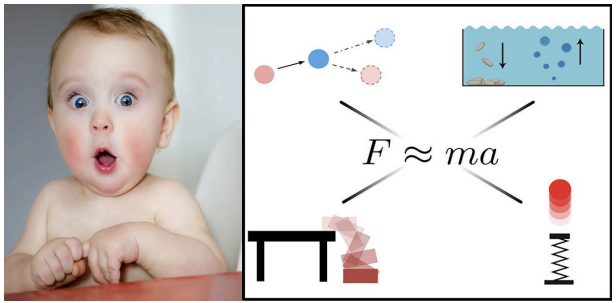


Outline

1. Motivation
2. Background
3. Bayesian Computer
4. Conclusion

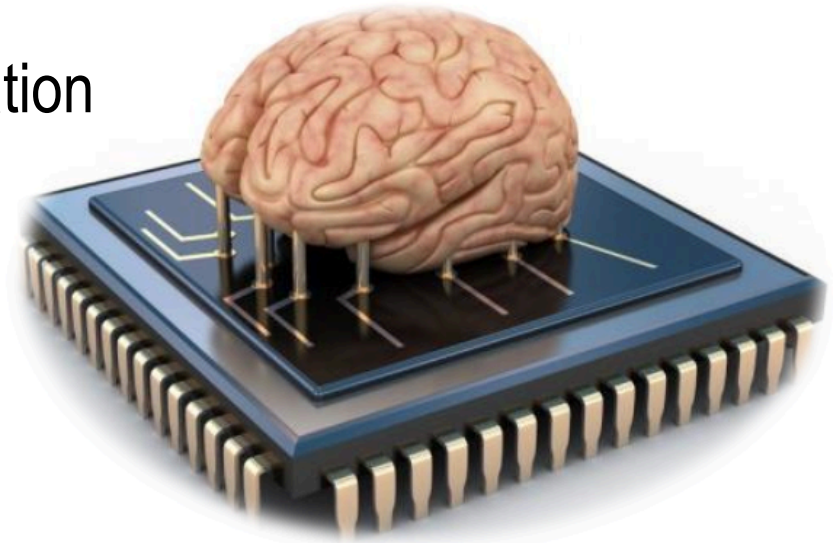
The Amazing Ability of Human Cognition

- Fact 1: “By the time we are six years old, we recognize more than 10^4 categories of objects” (Fei-Fei et al., PAMI, 2006)
  - Many categories per day
  - Kids can learn objects via only few positive examples
- Fact 2: Infants learn Newton’s Laws (Battaglia et al., PNAS, 2013)
  - To see is, famously, “to know what is where by looking”
- Fact 3: Responsiveness and power efficiency
  - Inference (e.g. Classification) in ~20ms + Motor control in ~400ms
  - At a power budget of 25W



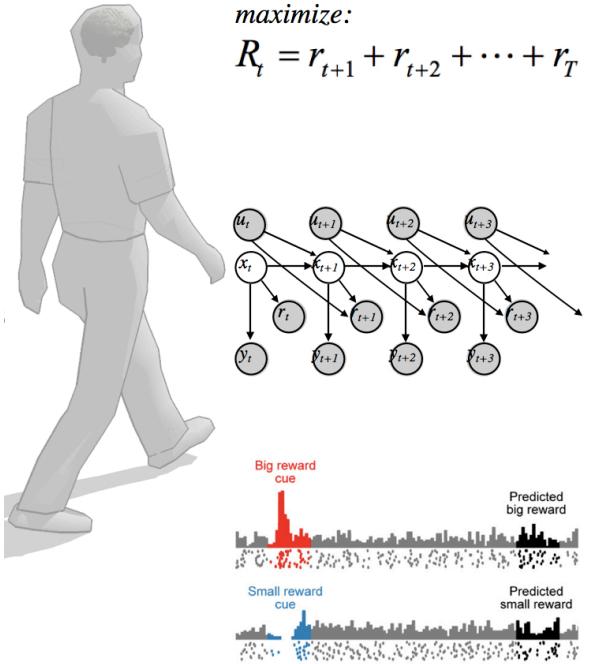
Problem: Can We Build Machines That Learn and Think like People

- Word-concept learning
- Category learning and generalization
- Physics learning
- Sensory-motor integration
- Mental simulation
- Theory learning
- …



Marr's Three Levels of Brain Analysis

- To understand the brain, we need three levels of knowledge
  - Computational
    - Describe and specify the problems we are faced with in a generic manner, but not how these problems are to be solved
  - Algorithmic
    - Representation and processing flow to perform computations
  - Implementational
    - Physical substrate or mechanism, and its organisation, in which computation is performed
    - Could be biological in the case of neurons and synapses, or in silicon using transistors, etc.

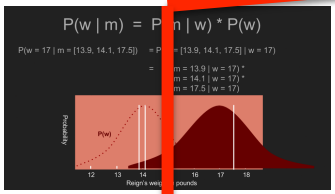


To Build Machines That Learn And Think Like People

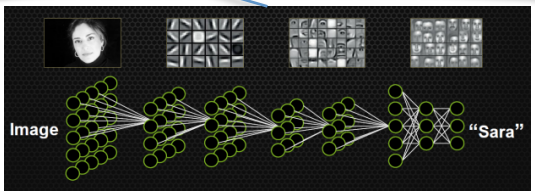
Computational Level

Has to stick to the human cognition problems

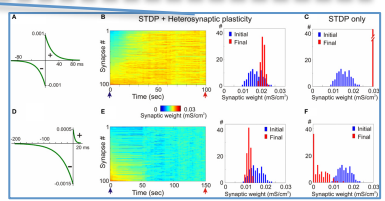
Algorithmic Level



Bayesian

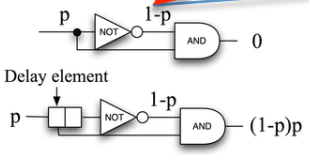


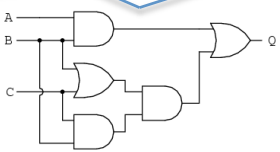
Deep Neural Network



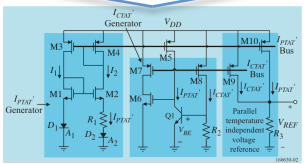
STDP Learning

Implementational Level

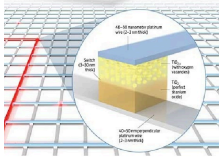




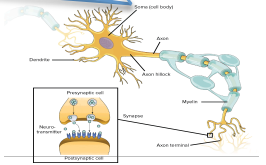
Digital circuits



Analog circuits



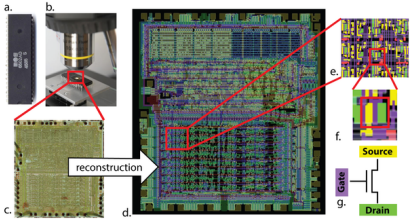
Memristor

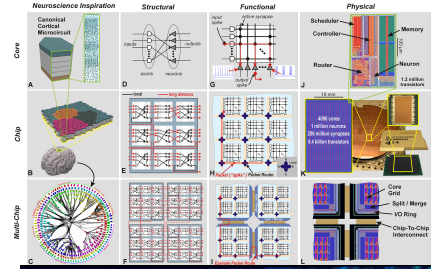


Neuron

Works Towards Learn And Think Like People

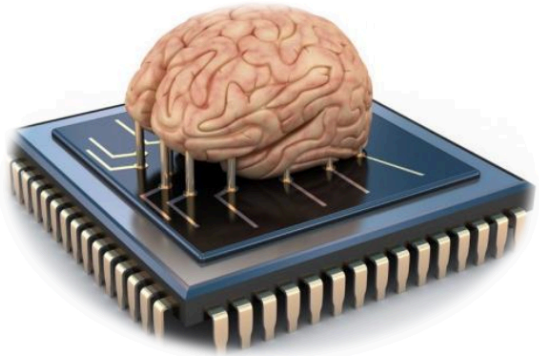
- Neuroimaging based techniques
  - Tons of papers (On almost every issue of Nature, Science, and Cell)
  - Problem: We cannot even understand how a CPU works by watching the patterns of electronic spikes (“Could a neuroscientist understand a microprocessor?” Eric Jonas & Konrad Kording, 2016)
- IBM TrueNorth (On-chip simulation of spiking neural network)
  - “A million spiking-neuron integrated circuit with a scalable communication network and interface.” Science’14
  - Problem: Biologically plausible but cannot prove to be psychosocially plausible
- BlueBrain (Brain simulation on supercomputer led by Henry Markram)
  - “Computer model of rat-brain.” Nature’15
  - “Reconstruction and Simulation of Neocortical Microcircuitry.” Cell’15
  - Problem: We cannot test any meaningful hypotheses with this brain model other than the vague “our rules produce similar activity”





What Do We Want To Build?

- A stochastic-by-nature Bayesian computer
  - Stochastic sampling circuit
    - True random number generator
    - Native Bayesian computing
    - Human-alike memory
      - Working memory, episodic memory, …
  - Programmed by probabilistic languages
    - e.g. STAN, Church, …
  - Cognition capability
    - Composability, generalization, and learning to learn



$$P(H \mid D) = \frac{P(H)P(D \mid H)}{P(D)}$$

Entropy

Outline

1. Motivation
2. Background
3. Bayesian Computer
4. Conclusion

What is Bayesian Theorem?

Prior probability *P(H)*: Probability of hypothesis *H* (knowledge about *H* is the correct hypothesis)

Likelyhood *P(D|H)*: Probability of observation *D* given hypothesis *h* holds

$$P(H \mid D) = \frac{P(H)P(D \mid H)}{P(D)}$$

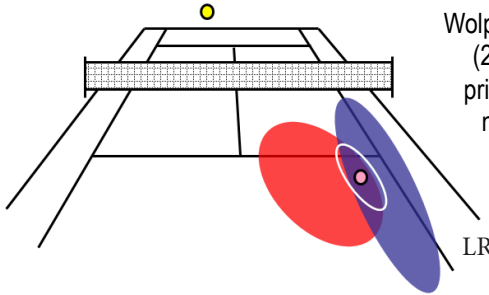
Posterior probability *P(H|D)*: Probability that hypothesis *h* holds given data (evidence) *D*

P(D): Marginal probability of data *D* (probability of observing D)

Why Bayesian Theorem matters?

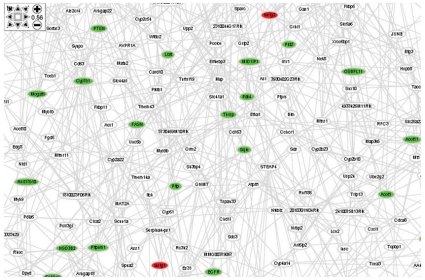
- Allows inference from **Prior** to **Posterior** under **uncertainly**
  - Priors come from all data external to the current study
- Decisions based on the likelihood ratio (LR) are statistically optimal
  - The firing rate of single neurons in the brain report evolving log LR values
- Bayesian inference uses the 'language' of probability to describe the complex relations among random variables

$$P(H \mid D) = \frac{P(H)P(D \mid H)}{P(D)}$$



Wolpert and Ghahramani (2000). Computational principles of movement neuroscience. Nature Neuroscience

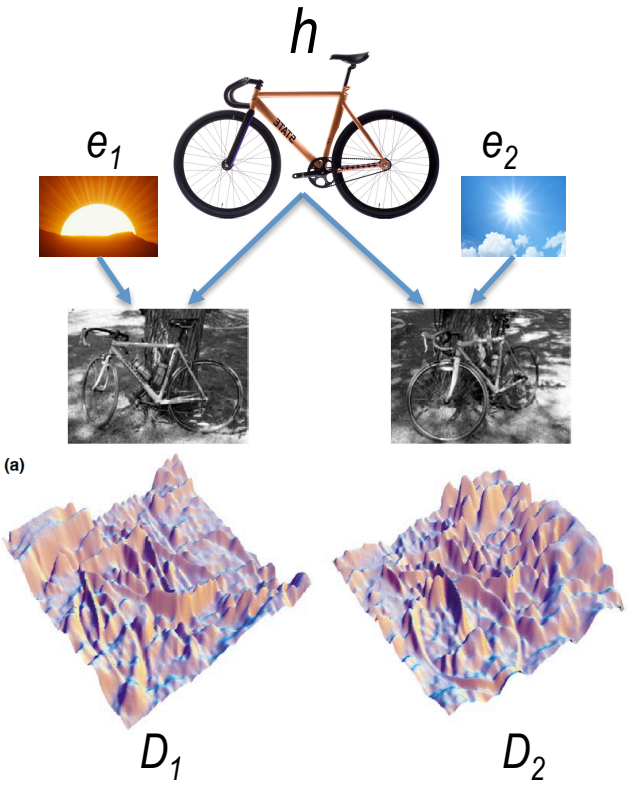
$$\mathrm{LR} = \frac{p(x_n = \mathrm{IN} \mid Y_n)}{p(x_n = \mathrm{OUT} \mid Y_n)}.$$



Yang et al. (2009). Validation of candidate causal genes for obesity that affect shared metabolic pathways and networks. Nature Genetics

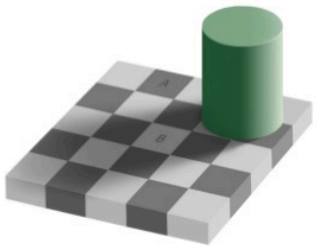
Bayesian Visual Perception

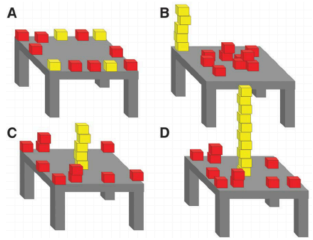
- The spectrum, *D*, of light wavelengths reflected from an object's surface into the observer's eye is a product of two spectra
  - The surface's color spectrum, *h*, and
  - The spectrum, *e*, of the light illuminating the scene
- Inferring the object's color given only the light reflected from it, under any conditions of illumination – is akin to solving P(*h*|D)
- This inference process can be formalized in a Bayesian framework (Brainard & Freeman, 1997)

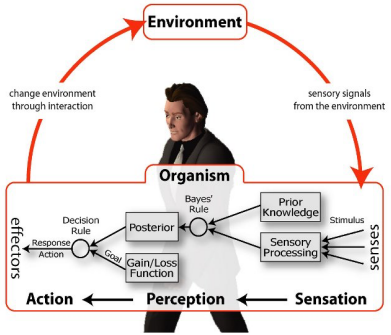


More Evidence

- Object perception as Bayesian inference (Kersten et al. Annual Review of Psychology'04)
- Infants consider both the sample and the sampling process in inductive generalization (Gweon et al. PNAS'10)
- Pure reasoning in 12-month-old infants as probabilistic inference (Teglas et al. Science'11)
- Human-level concept learning through probabilistic program induction (Lake et al., Science'15)
- …



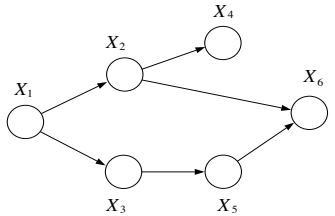


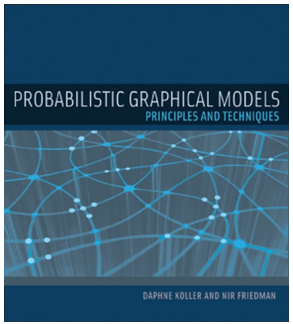


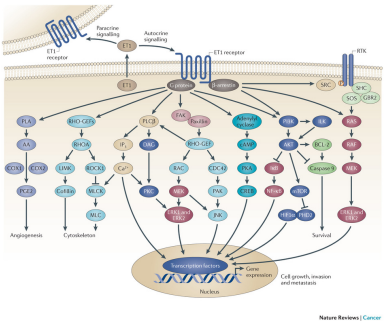
Bayesian Network in Machine Learning

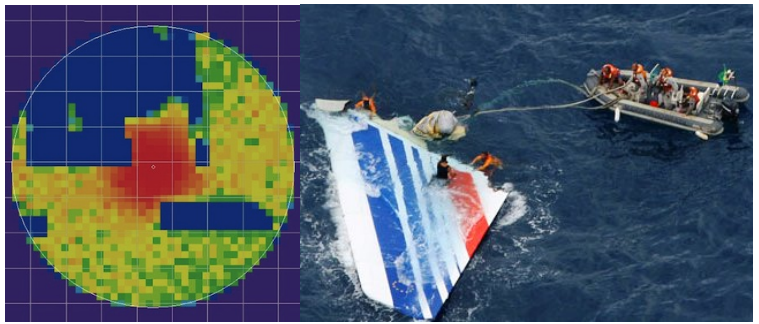
- Graphical model/Bayesian network/structure learning/latent variable
  - E.g. P(x1, x2, x3, x4, x5, x6)

    =p(x1)p(x2|x1)p(x3|x1)p(x4|x2) p(x5|x3) p(x6|x2,x5)
  - How do we build a Bayesian network?
    - Markov Chain Monte Carlo









Bayesian network helps find Air France Flight 447 debris

Outline

1. Motivation
2. Background
3. Bayesian Computer
4. Conclusion

What to compute?

- Inference, i.e., learning the posterior probability
  - Calculate characteristics of a complicated multivariate probability distribution

$$P(h_1, h_2, \ldots, h_n|D) = \frac{P(D|h_1, h_2, \ldots, h_n)P(h_1, h_2, \ldots, h_n)}{P(D)}$$

- Approach 1: Monte Carlo Markov Chain (MCMC) simulation
  - Simulate k observations $X^{(1)}$, $X^{(2)}$, …, $X^{(k)}$ to estimate the characteristics of interest
  - Metropolis-Hastings, Gibbs, Importance sampling, Slice sampling, …
- Approach 2: Variational Bayesian methods
  - "Provides a locally-optimal, exact analytical solution to an approximation of the posterior"

MCMC Example : Metropolis-Hastings Algorithm

Algorithm 5.1: Metropolis-Hastings

Starting with $\mathbf{X}^{(0)} := (X_1^{(0)}, \ldots, X_p^{(0)})$ iterate for $t = 1, 2, \ldots$

1. Draw $\mathbf{X} \sim q(\cdot|\mathbf{X}^{(t-1)})$.
2. Compute

$$\alpha(\mathbf{X}|\mathbf{X}^{(t-1)}) = \min\left\{1, \frac{f(\mathbf{X}) \cdot q(\mathbf{X}^{(t-1)}|\mathbf{X})}{f(\mathbf{X}^{(t-1)}) \cdot q(\mathbf{X}|\mathbf{X}^{(t-1)})}\right\}.$$

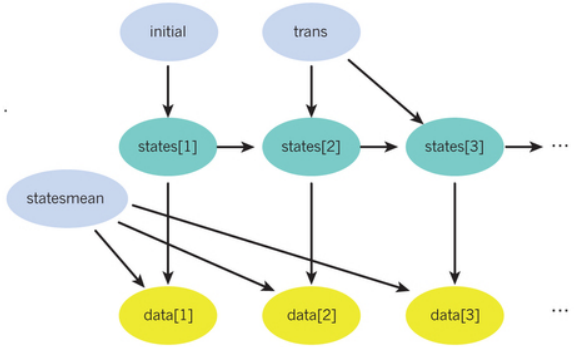
3. With probability $\alpha(\mathbf{X}|\mathbf{X}^{(t-1)})$ set $\mathbf{X}^{(t)} = \mathbf{X}$, otherwise set $\mathbf{X}^{(t)} = \mathbf{X}^{(t-1)}$.

2. Sample a potentially highly complex function (problem-specific)

3. Has to iterate for a huge number, i.e. a long Markov chain

1. Stochastic computation by nature -> Inherently inefficient to program with conventional programming language and to execute on current processors

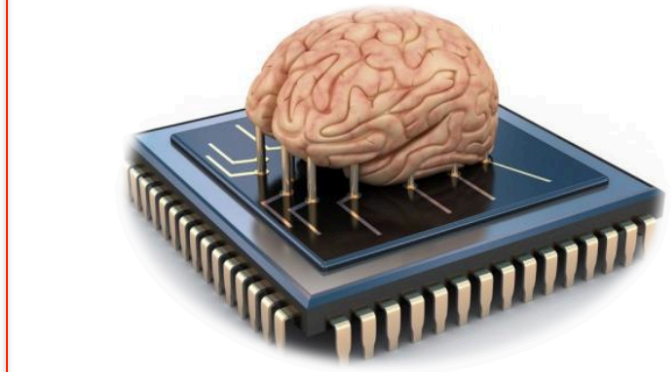
Problem 1: Probabilistic Inference on Stochastic Hardware

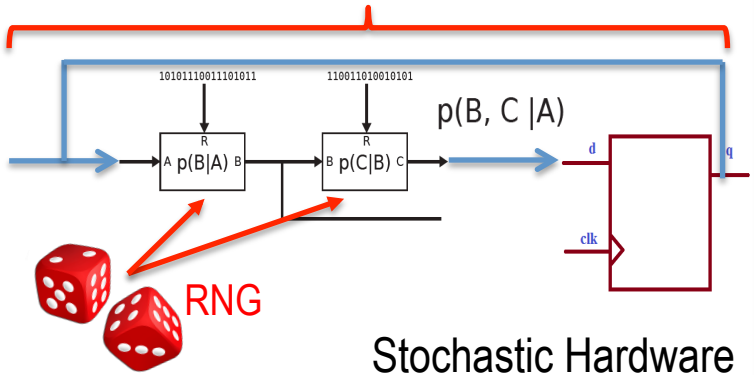



```
statesmean = [-1, 1, 0]  # Emission parameters.
initial    = Categorical([1.0/3, 1.0/3, 1.0/3]) # Prob distr of state[1].
trans      = [Categorical([0.1, 0.5, 0.4]), Categorical([0.2, 0.2, 0.6]),
              Categorical([0.15, 0.15, 0.7])]   # Trans distr for each state.
data       = [Nil, 0.9, 0.8, 0.7, 0, -0.025, -5, -2, -0.1, 0, 0.13]

@model hmm begin # Define a model hmm.
 states = Array(Int, length(data))
 @assume(states[1] ~ initial)
 for i = 2:length(data)
   @assume(states[i] ~ trans[states[i-1]])
   @observe(data[i]  ~ Normal(statesmean[states[i]], 0.4))
 end
 @predict states
end
```

Probabilistic Program

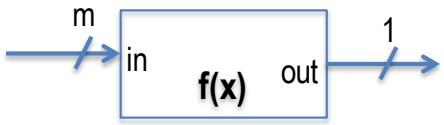




Stochastic Hardware

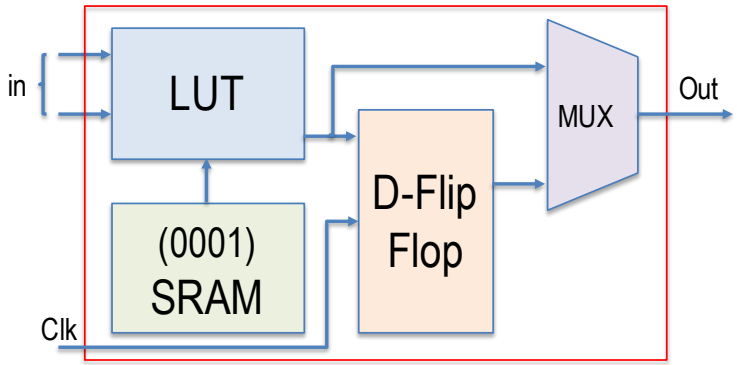
Problem 2: Sampling

- A Boolean logic gate (LUT here) is characterized by its truth table

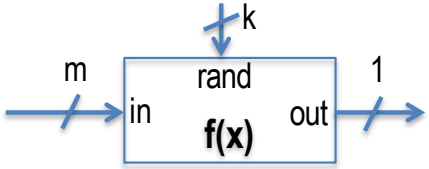


in	out
00	0
01	0
10	0
11	1

f(in) = AND(in1, in2)



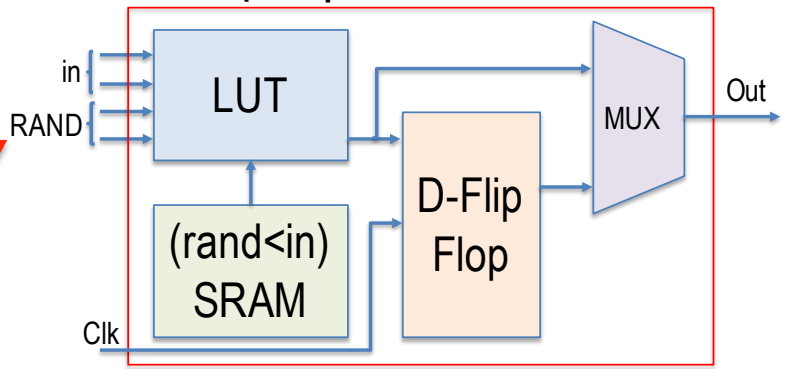
- A stochastic gate is characterized by its conditional probability table



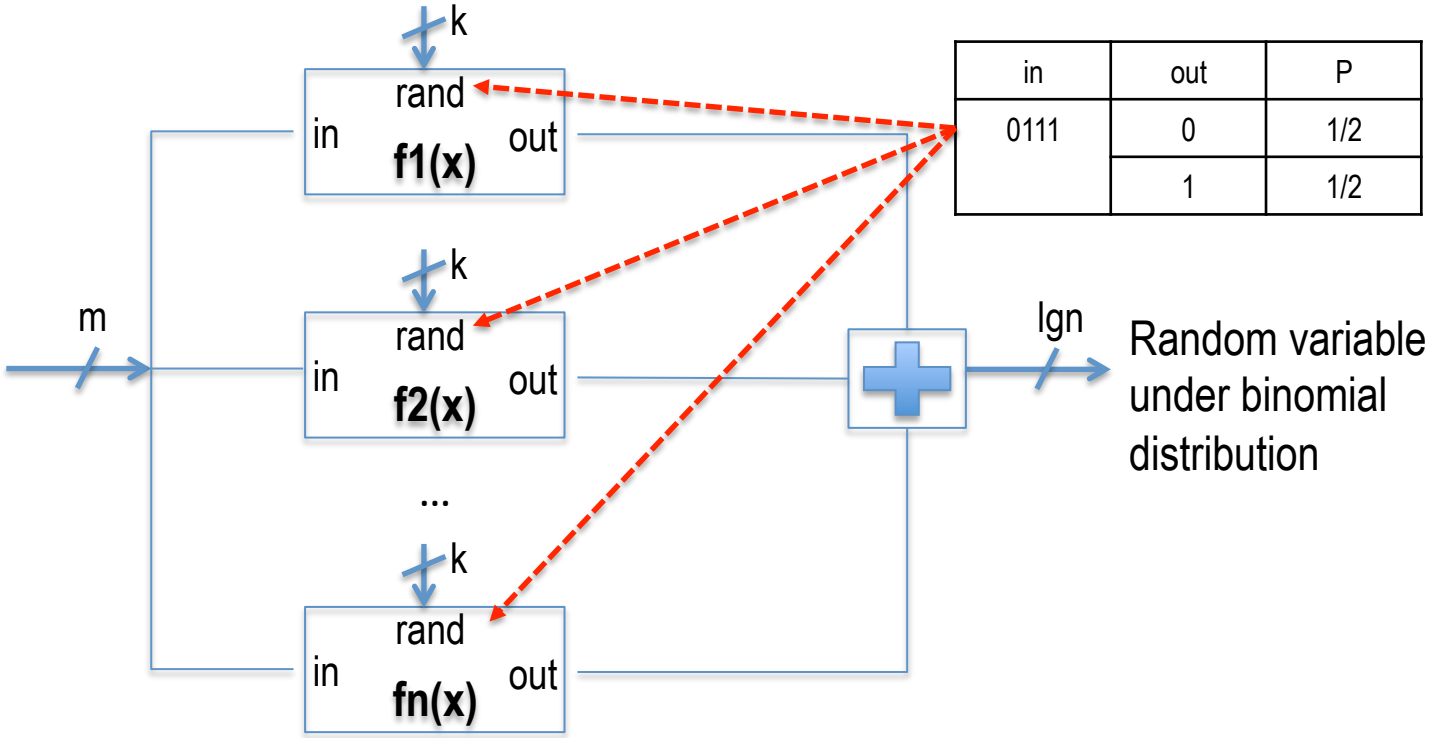
RAND	out	P
0000	0	1
	1	0
…	…	…
0111	0	1/2
	1	1/2

f(in) = rand < in i.e., P(out | in)

Source of entropy



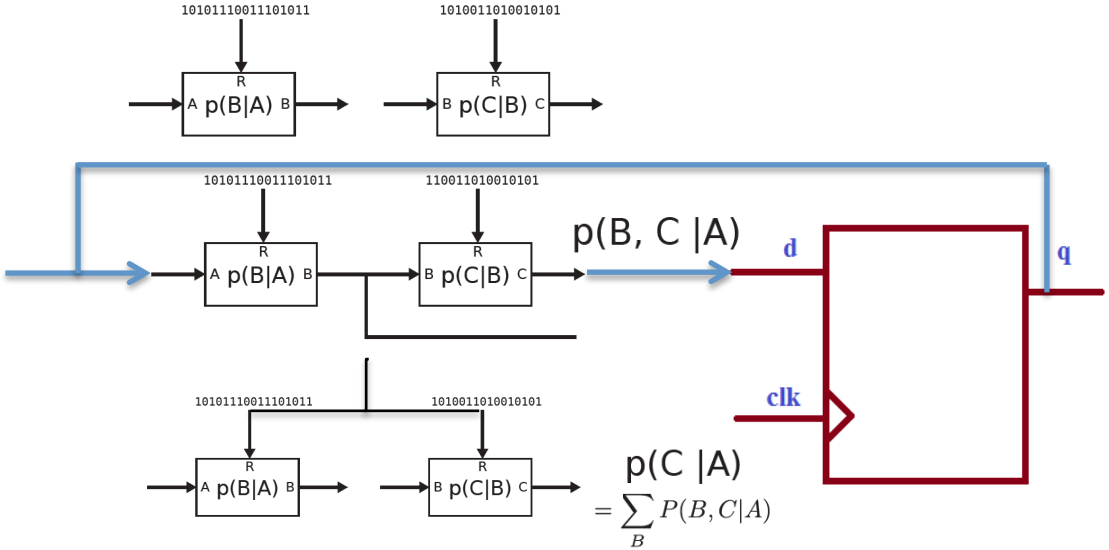
An Example Combinatorial Stochastic Circuit: Binomial Distribution



in	out	P
0111	0	1/2
	1	1/2

Sequential Stochastic Circuit

- An MCMC process can be implemented with a finite sequence of stochastic gates and registers



Problem 3: Parallelization

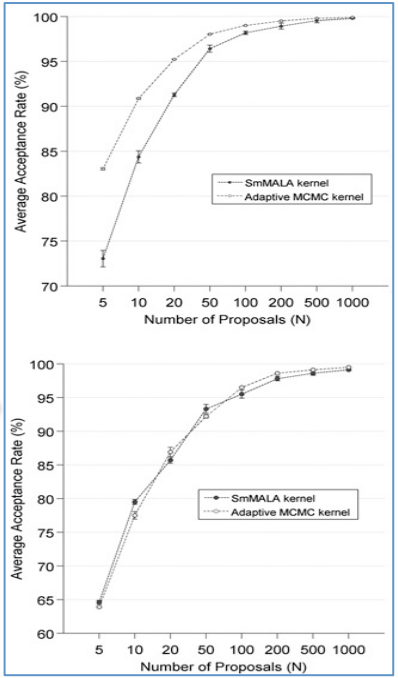
Algorithm 5.1: Metropolis-Hastings

Starting with $\mathbf{X}^{(0)} := (X_1^{(0)}, \ldots, X_p^{(0)})$ iterate for $t = 1, 2, \ldots$

1. Draw $\mathbf{X} \sim q(\cdot|\mathbf{X}^{(t-1)})$.
2. Compute

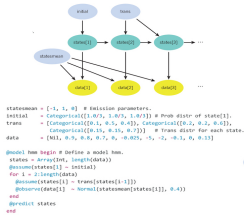
$$\alpha(\mathbf{X}|\mathbf{X}^{(t-1)}) = \min\left\{1, \frac{f(\mathbf{X}) \cdot q(\mathbf{X}^{(t-1)}|\mathbf{X})}{f(\mathbf{X}^{(t-1)}) \cdot q(\mathbf{X}|\mathbf{X}^{(t-1)})}\right\}.$$

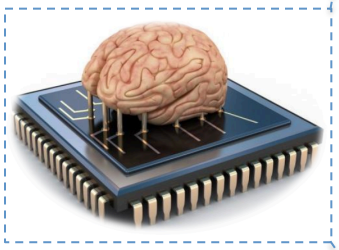
3. With probability $\alpha(\mathbf{X}|\mathbf{X}^{(t-1)})$ set $\mathbf{X}^{(t)} = \mathbf{X}$, otherwise set $\mathbf{X}^{(t)} = \mathbf{X}^{(t-1)}$.



*Calderhead, A general construction for parallelizing Metropolis−Hastings algorithms, PNAS, 2014

Overall Microarchitecture





Bayesian Computer

...

Stochastic Multiprocessor n

Stochastic Multiprocessor 1

...

Stochastic Multiprocessor 0

Program Fetch

Program Mapper

Programmable Controller

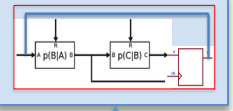
Global Scheduler

True Random Number Generator

Sampling Unit

Sampling Unit

...



Memory Interface

Register File

Register File

...

Register File

Human Memory Model

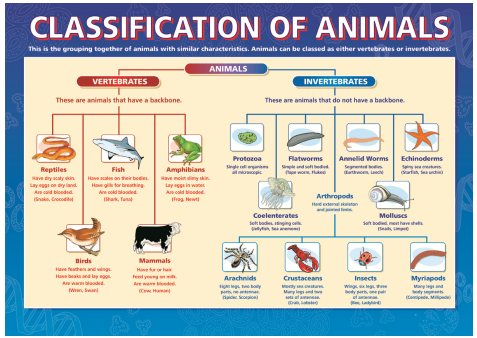
Local Exemplar Memory

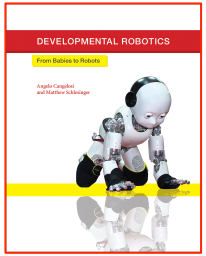
Shared Memory

Applications

- Ongoing
  - Concept learning
  - Visual category learning
  - Human memory via exemplar model
  - Physics learning
- Next
  - Visual Turing test
  - Developmental robot

(Mining Wiki?)





Outline

1. Motivation
2. Background
3. Bayesian Computer
4. Conclusion

Conclusion

- Objective
  - Building machines that Learn and Think like People
- Approach
  - Probabilistic computing for cognition
  - Native Bayesian computing on FPGA
  - Parallel MCMC
- Bayeisan based learning and inference to make a “better” human being?